# Enhancing Video Streaming Delivery over Wired/Wireless Networks

Panagiotis Papadimitriou, Vassilis Tsaoussidis, and Chi Zhang

***Abstract*—Delivering streaming video over wired/wireless networks is challenging, since link errors commonly compromise throughput performance, smoothness, and eventually impair the perceptual video quality. We combine an efficient *Additive Increase Multiplicative Decrease* (*AIMD*) transport protocol with an end-to-end mechanism that effectively decouples wireless from congestion loss to abolish the damage of error-induced multiplicative decrease on flow throughput and smoothness. Based on simulation results, we show that our combined approach provides the desired functionality to bind operationally wired and wireless links, within the framework of bandwidth efficiency, smoothness and fairness. The proposed mechanism can be easily adapted and incorporated into existing AIMD protocols, allowing them to utilize more efficiently wireless resources.**

***Index Terms*—transport protocols, wireless networks, QoS, video streaming.**

## I. Introduction

An increasing demand for multimedia data delivery coupled with reliance on best-effort networks, such as the Internet, has spurred interest in efficient transport solutions for multimedia streams. Video streaming, in particular, is comparatively intolerant to delay and variations of throughput and delay. Unlike bulk-data transfers, video delivery requires a minimum and continuous bandwidth guarantee. It is also affected by reliability factors, such as packet drops due to congestion or link errors. In general, streaming applications yield satisfactory performance only under certain *Quality of Service* (*QoS*) provisions, which may vary depending on the application task and the type of media involved.

Today's multimedia applications are expected to run in physically heterogeneous environments composed of both wired and wireless components. Wireless links exhibit distinct characteristics, such as limited bandwidth, varying error-rates and potential handoff operations. Consequently, QoS requirements in wireless networking are stringent and complicated, taking additionally into account the influencing mobile device characteristics and limitations. For example, a considerable number of mobile devices offer limited buffer capacities, being unable to smooth the fluctuations in the receiving rate. In this case, the task of smooth delivery is primarily delegated to the transport protocol.

P. Papadimitriou and V. Tsaoussidis are with the Department of Electrical and Computer Engineering, Demokritos University, Xanthi, 67100, Greece (e-mail: ppapadim@ee.duth.gr; vtsaousi@ee.duth.gr).

Chi Zhang is with Juniper Networks, Sunnyvale, CA, USA (e-mail: chizhang@juniper.net).

*Transmission Control Protocol* (*TCP*) is basically designed to provide a reliable service for wired Internet. The *Additive Increase Multiplicative Decrease* (*AIMD*) algorithm [5], incorporated into standard TCP versions, achieves stability and converges to fairness when the demand of competing flows exceeds the channel bandwidth. However, most existing TCP mechanisms do not satisfy the need for universal functionality in heterogeneous wired/wireless environments, since they do not flexibly adjust the rate and pattern of the transmitted multimedia streams to the characteristics of the end-to-end network path. Authors in [20] outline three major shortfalls of TCP: (i) ineffective bandwidth utilization, (ii) unnecessary congestion-oriented responses to wireless link errors (e.g. fading channels) and operations (e.g. handoffs), and (iii) wasteful window adjustments over asymmetric, low-bandwidth reverse paths. Furthermore, TCP's process of probing for bandwidth and reacting to observed congestion causes oscillations to the achievable transmission rate. TCP may also introduce arbitrary delays, since it enforces reliability and in-order delivery. In response to standard TCP limitations, several TCP protocol extensions [2, 8, 22, 23] have emerged providing more effective bandwidth utilization and sophisticated mechanisms for congestion control.

*User Datagram Protocol* (*UDP*) has been widely used instead of TCP by media-streaming applications. UDP lacks all basic mechanisms for error recovery and flow/congestion control. Thus, it allows for transmission attempts at application speed. That said, UDP can not guarantee reliability, and certainly is not able to deal with network delays either. In [14] we have shown that UDP may perform worse than TCP in several occasions.

Since TCP is not preferred by multimedia applications and UDP poses a threat to network stability, rate-based congestion control has become an attractive alternative. Rate-based mechanisms directly control the transmission rate of the connection, based on either measurements taken at the end host or feedback from the network [8, 13, 17]. Avoiding the burstiness occasionally induced by the window-based mechanisms, rate-based protocols generate a smooth data flow by spreading the data transmission across a time interval. Therefore, rate-based mechanisms compose plausible candidates when smooth delivery composes a primary objective. However, the challenge does not lie in simply achieving smoothness, but rather providing adequate efficiency and resilience to the inherent characteristics of wireless links. More precisely, a suitable protocol for wired/wireless networks should be able to detect the nature of the errors that result in packet loss in order to determine the appropriate error-recovery strategy. Based on such an approach, the sender would not be obliged to reduce its transmission rate in the event of a wireless error or handoff.

In contrast to transport-layer solutions, a series of independent mechanisms have been proposed, which normally interact with the transport protocol and provide reliable transmission over wireless links [1, 6, 11]. Most of them operate on link-layer. However, link-layer approaches may degrade performance, especially in the presence of highly variable error rates. Local error recovery may alter the characteristics of the network affecting the functionality of higher layer protocols. For example, local retransmission could result in packet reordering or in large fluctuations of *Round Trip Time* (*RTT*). In addition, concurrent responses from both local and end-to-end error control may result in undesirable interactions, causing inefficiencies and potentially instability. Considering real-time traffic where data packets bear information with a limited useful lifetime, retransmissions are often a wasted effort. In such conditions, unfruitful retransmissions deliver delayed packets which are either discarded, or at the worst they obstruct the proper reconstruction of oncoming packets.

Our objective is to combine an efficient transport protocol with a mechanism that provides robustness and resilience to transient errors, achieving uninterrupted and smooth video delivery in wired/wireless environments. In this context, we propose an end-to-end mechanism that effectively decouples wireless from congestive loss to abolish the damage of error-induced multiplicative decrease on flow throughput and smoothness. This mechanism can be easily adapted and incorporated into existing transport-layer solutions, spanning from TCP to rate-based AIMD protocols. Since we focus on enhancing streaming video delivery, we incorporate the proposed mechanism into *Scalable Streaming Video Protocol* (*SSVP*) [13, 15] and evaluate its efficiency in terms of bandwidth utilization, smoothness and fairness. SSVP is an AIMD-oriented rate control scheme optimized for video streaming applications.

We organize the rest of the paper, as follows. The following section summarizes related work. In Section 3 we describe the proposed loss differentiation mechanism and incorporate it into SSVP. Section 4 includes conclusive performance studies based on simulations. Finally, in the last section we highlight our conclusions.

## II. Related Work

In the sequel, we provide a taxonomy of the most prominent approaches that tackle the limitations of media delivery over wired/wireless networks. In this context, we discuss separately end-to-end enhancements for heterogeneous environments, transport-layer solutions for efficient media delivery and selected mechanisms operating on the link layer.

### A. End-to-end Enhancements for Wireless Links

Numerous proposals have been presented in order to improve transport-layer efficiency over wireless links [1, 11]. Most related research efforts focus on bulk-data transmission, and are usually pronounced as enhanced TCP versions. *TCP Probing* [19] grafts a probing cycle and an *Immediate Recovery Strategy* into standard TCP in order to control effectively the throughput/overhead trade-off. *Freeze-TCP* [10] distinguishes handoffs from congestion through the use of the advertised window. *WTCP* [18] implements a rate-based congestion control replacing entirely the ACK-clocking mechanism. Selected end-to-end loss differentiation algorithms are applied to *TCP-friendly Rate Control* (*TFRC*) and their efficiency is analyzed in [4]. Authors in [3] present and analyze the bandwidth estimation schemes implemented at the sender side of a TCP connection. In addition, they propose *TIBET* (*Time Intervals based Bandwidth Estimation Technique*), a new bandwidth estimation scheme implemented within the TCP congestion control procedure, which enhances TCP performance over wireless links.

### B. Transport-layer Approaches for Multimedia Traffic

The literature includes several proposals towards efficient rate control for media-streaming applications in the Internet. *Rate Adaptation Protocol* (*RAP*) [17] is a rate-based protocol which employs an AIMD algorithm for the transmission of real-time streams. The sending rate is continuously adjusted by RAP in a TCP-friendly fashion by using feedback from the receiver. However, since RAP employs TCP's congestion control parameters (1, 0.5), it causes short-term rate oscillations, primarily due to the multiplicative decrease. *Datagram Congestion Control Protocol* (*DCCP*) [12] is a new transport protocol that provides a congestion-controlled flow of unreliable datagrams. DCCP is intended for time-sensitive applications which have relaxed packet loss requirements. The protocol provides the application with a choice of congestion control mechanisms via *Congestion Control IDs* (*CCIDs*), which explicitly name standardized congestion control mechanisms (i.e. TCP-like and TFRC).

Since TCP is rarely chosen to transport multimedia traffic over the Internet, *TCP-friendly* protocols [8, 22, 23] constitute an elegant framework for multimedia applications. We consider as TCP-friendly any protocol whose long-term arrival rate does not exceed the one of any conformant TCP in the same circumstances [7]. TCP-friendly congestion control maintains network stability by promptly responding to congestion and is also cooperative with other flows, while it commonly provides more efficient QoS (e.g. smoothed sending rate and bounded latency for multimedia applications). TFRC [8] is a representative TCP-friendly protocol, which adjusts its transmission rate in response to the level of congestion, as estimated based on the calculated loss rate. Multiple packet drops in the same RTT are considered as a single loss event by TFRC and hence, the protocol follows a more gentle congestion control strategy. More precisely, the TFRC sender uses the following response function:

$$T(p, RTT, RTO) = \frac{1}{RTT\sqrt{\frac{2p}{3}} + RTO\ (3\sqrt{\frac{3p}{8}})\ p\ (1 + 32p^2)} \qquad (1)$$

where $p$ is the steady-state loss event rate and $RTO$ is the retransmission timeout value. Equation (1) enforces an upper bound on the sending rate $T$. However, the throughput model is quite sensitive to parameters (i.e. p, RTT), which are often difficult to measure efficiently and to predict accurately. Also, the long-term TCP throughput equation does not capture the transit and short-lived TCP behaviors, and it is less responsive to short-term network and session dynamics [21].

*GAIMD* [23] is a TCP-friendly protocol that generalizes AIMD congestion control by parameterizing the additive increase rate $\alpha$ and multiplicative decrease ratio $\beta$. For the family of AIMD protocols, authors in [23] derive a simple relationship between $\alpha$ and $\beta$ in order to be friendly to standard TCP:

$$\alpha = \frac{4\,(1-\beta^2)}{3} \tag{2}$$

Based on experiments, they propose an adjustment of $\beta = 0.875$ as an appropriate smooth decrease ratio, and a moderated increase value $\alpha = 0.31$ to achieve TCP friendliness.

### C. Link-layer Enhancements

There are several techniques operating on the link layer, which attempt to ameliorate the impact of wireless errors [1, 11]. The most remarkable implementations, which provide error-correction, are *Forward Error Correction* (*FEC*) and *Automatic Repeat Request* (*ARQ*) [6]. FEC introduces added overhead to data bits in order to cope with data corruption. Corrupted packets are directly corrected, without retransmission, which is critical for lossy links exhibiting long delays. However, the redundant information is not exploited in the absence of link errors resulting in a waste of bandwidth. Furthermore, FEC requires additional resources in CPU processing time, memory and power consumption.

On the other hand, ARQ mechanisms are invoked when packets containing bit errors can not be corrected. In such case, the erroneous packets are discarded and a retransmission is directly triggered. Unlike FEC, ARQ allocates additional network resources only when a packet is retransmitted. The mechanism generally operates more efficiently for low bit rates. An undesirable effect of ARQ is that it may interfere with the transport protocol [1].

## III. SSVP with Loss Differentiation

SSVP employs a receiver-centric congestion control mechanism, which does not rely on QoS functionality in routers, such as *Random Early Detection* (*RED*) [9], *Explicit Congestion Notification* (*ECN*) [16] or other *Active Queue Management* (*AQM*) mechanisms. The protocol, in a complementary role, operates on top of UDP, relying on sender and receiver interaction. The recipient uses control packets to send feedback of reception statistics to the sender. In accordance with the relaxed packet loss requirements of streaming video and considering the delays induced by retransmitted packets, SSVP does not integrate reliability into UDP datagrams. Hence, control packets do not trigger retransmissions. However, they are effectively used in order to determine bandwidth and RTT estimates, and properly adjust the rate of the transmitted video stream. The recipient uses packet drops or re-ordering as congestion indicator. SSVP employs an AIMD-oriented congestion control mechanism with $\alpha = 0.2$ and $\beta = 0.875$. The transmission rate is controlled by properly adjusting IPG. If no congestion is sensed, IPG is reduced additively; otherwise, it is increased multiplicatively. Further details in the operation of SSVP can be found in [13, 15].

In the sequel, we propose a *Loss Discrimination Algorithm* (*LDA*), which allows SSVP to distinguish wireless from congestion loss. The proposed LDA is a pure end-to-end mechanism and does not require any modifications in the network infrastructure or the underlying network protocol. It relies on RTT estimation to measure current network conditions and effectively adapt to the dynamics of the network.

Consider an SSVP source that transmits $n$ packets with packet lengths $S_1$, $S_2$, …, $S_n$ during a time period $t$, where $S_i$ represents the $i^{th}$ packet. The average throughput used by the connection is given by:

$$\text{throughput} = \frac{1}{t}\sum_{i=1}^{n} S_i = \frac{n\overline{S}}{t} \tag{3}$$

where $\overline{S}$ denotes the average packet length. SSVP performs rate adjustments per-RTT in order to achieve a relatively responsive behavior to the sudden changes of bandwidth availability. In this context, we define $K$ as a function of IPG:

$$K_j = \frac{RTT_j}{IPG_j} \tag{4}$$

representing the number of packets transmitted within $j^{th}$ RTT. Assuming a measurement period of one RTT, flow throughput is given by:

$$\text{throughput} = \frac{K_j\,\overline{S}}{RTT_j} \tag{5}$$

Combining equations (4) and (5), we obtain the instantaneous transmission rate $R_i$ for the SSVP flow:

$$R_i = \frac{\overline{S}}{IPG_i} \tag{6}$$

SSVP does not inherently incorporate any loss differentiation mechanism, and therefore the protocol invokes congestion-oriented responses to all wireless errors. Therefore, upon detecting packet loss, IPG is increased multiplicatively:

$$IPG_{i+1} = \frac{IPG_i}{\beta} \tag{7}$$

Apparently, an increase in the IPG directly affects the transmission rate, as well as flow throughput. Since the protocol does not decouple wireless loss from congestion, wireless errors result in considerable throughput degradation.

We consider a typical scenario for streaming video delivery across a network path that includes a wireless link. As

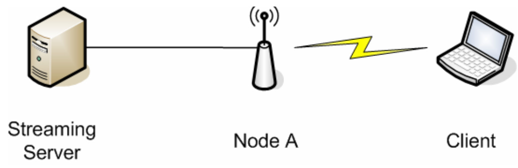



**Fig. 1.** Typical Wireless Scenario

depicted in Fig. 1, a streaming server transmits data to a receiver located in the wireless network. In the event of packet loss, the proposed LDA virtually suffices to observe current RTT and subsequently determine the nature of the loss. Let $RTT_{min}$ denote the round-trip propagation delay and *qdelay* the total queuing delay in all buffers across the network path. As soon as the bottleneck channel has been fully utilized, a queue is being built up and RTT is currently:

$$\mathrm{RTT} = \mathrm{RTT_{min}} + \mathrm{qdelay} \tag{8}$$

In the situation where the queue occupies the whole bottleneck buffer, RTT is eventually maximized and expressed as:

$$\mathrm{RTT_{max}} = \mathrm{RTT_{min}} + \mathrm{qdelay_{max}} \tag{9}$$

The LDA interacts with the protocol monitoring $\mathrm{RTT_{min}}$ and $\mathrm{RTT_{max}}$, while the queuing delay can be derived by deducting $\mathrm{RTT_{min}}$ from the last RTT measured. In the absence of wireless loss, $\mathrm{RTT_{max}}$ is normally observed before congestion control is triggered. Practically, upon packet loss if the last RTT is close to the $\mathrm{RTT_{min}}$, the bottleneck is not congested and the loss is due to a link error. On the other hand, a measured RTT substantially larger than $\mathrm{RTT_{min}}$ and close to $\mathrm{RTT_{max}}$ indicates a congestive loss. Following these observations, the protocol's congestion control is complemented with the following loss differentiation algorithm: upon the detection of packet loss the transmission rate is decreased multiplicatively (i.e. via the multiplicative increase of IPG) with the standard decrease ratio $\beta$, only when the following condition is satisfied:

$$\frac{\mathrm{qdelay}}{\mathrm{qdelay_{max}}} = \frac{\mathrm{RTT} - \mathrm{RTT_{min}}}{\mathrm{RTT_{max}} - \mathrm{RTT_{min}}} \geq \mathrm{qthresh} \tag{10}$$

Threshold *qthress* in equation (10) specifies the point of queue length where packet loss is considered to be congestion-induced. The specific threshold is adjusted experimentally at 0.5 for SSVP. Hence, when the queue occupies less than half of the buffer size, packet drops do not trigger congestion control invocations. Certainly, *qthress* can be adjusted differently in order to modify the protocol's error recovery strategy.

## IV. Performance Evaluation

### A. Experimental Environment

The evaluation plan was implemented on the *NS-2* network simulator. Simulations were conducted on a single-bottleneck *dumbbell* topology (Fig. 2) with a bottleneck capacity of 10 Mbps and a round-trip link delay of 64 ms. The bottleneck link is shared by competing MPEG and FTP connections. The capacity of all access links to the sink nodes is set to 1 Mbps. NS-2 error models were inserted into the access links to the MPEG sink nodes. The error models were configured on both directions of the link traffic. We used the Bernoulli model in order to simulate the link errors with packet error rate (*PER*) adjusted at 0.02, unless otherwise explicitly stated. The routers are drop-tail with buffer size adjusted in accordance with the *bandwidth-delay* product. We set the packet size to 1000 bytes for all system flows and the maximum congestion window to 64 KB for all TCP connections. The duration of each simulation is 60 seconds. All the results are collected after 2 sec in order to avoid the skew introduced by the warming up effect.

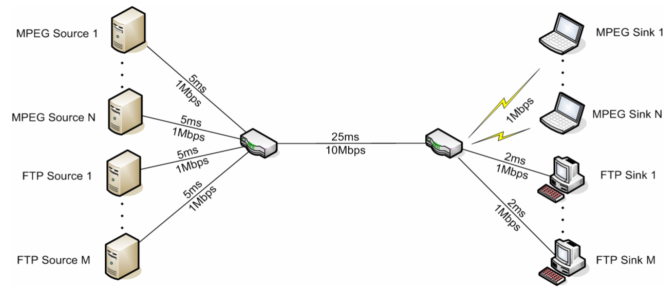



**Fig. 2.** Simulation Topology

In order to simulate MPEG traffic, we developed an *MPEG-4 Traffic Generator*. The traffic generated closely matches the statistical characteristics of an original MPEG-4 video trace. MPEG-4 coding standard is based on I (*intra* picture), P (*predictive*) and B (*bidirectional*) frames. The compression initiates by encoding a single I frame, followed by a group of P and B frames. P frames carry the signal difference between the previous frame and motion vectors, while B frames are interpolated; the encoding is based on the previous and the next frame. The model developed is based on *Transform Expand Sample* (*TES*). We used three separate *TES* models for modeling I, P, and B frames, respectively. The resulting MPEG-4 stream is generated by interleaving data obtained by the three models.

We hereby refer to the performance metrics supported by our simulation model. Since the simulation topology includes competing MPEG and FTP connections, our performance metrics are applied separately to the MPEG and FTP traffic. *Throughput* is used to measure the efficiency in link utilization. Long-term fairness is measured by the *Fairness Index*, derived from the formula given in [5], and defined as:

$$\text{Fairness Index} = \frac{\left(\sum_{i=1}^{n} \mathrm{Throughput}_i\right)^2}{n\left(\sum_{i=1}^{n} \mathrm{Throughput}_i^{\,2}\right)}$$

where $Throughput_i$ is the throughput of the $i^{th}$ flow and $n$ is the total number of flows. In order to quantify the performance on video delivery, we monitor packet inter-arrival times and eventually distinguish the packets that can be effectively used by the client application from delayed packets (according to a configurable packet inter-arrival threshold). The proportion of delayed packets is demoted as *Delayed Packets Rate*. In accordance with video streaming requirements, we adjusted the packet inter-arrival threshold at 75 ms.

### B. Results and Discussion

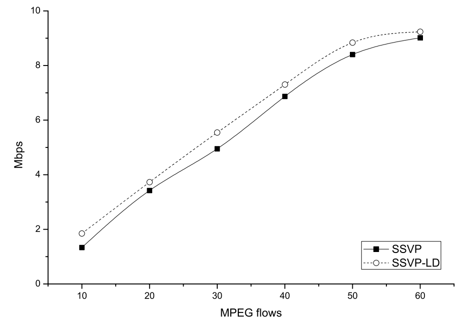
In the sequel, we demonstrate conclusive performance studies based on selected simulation results. Initially, we assess the efficiency of the proposed LDA in terms of link utilization, fairness and smooth video delivery. In this context, we simulated a wide range of MPEG flows (10-60) over (i) SSVP, and (ii) SSVP with LDA (SSVP-LD), successively. We measured *System Throughput and Fairness Index,* and we additionally demonstrate statistics from delayed packets which compose an influencing factor for perceived video quality (Fig. 3).

Throughput performance (Fig. 3a) reflects the beneficial role of the integrated LDA. SSVP-LD is relatively immunized by the link errors across the wireless channel and utilizes a higher fraction of the available bandwidth, regardless of link multiplexing. On the other hand, standard SSVP invokes congestion-oriented responses to the wireless errors, diminishing the throughput rate. We note that the protocol incorporates a gentle decrease ratio (i.e. $\beta = 0.875$), and therefore the impact of an unnecessary multiplicative decrease is not destructive on throughput performance. Indeed, a protocol with conventional congestion control parameters (i.e. $\alpha = 1$, $\beta = 0.5$), such as TCP, would experience significant throughput degradation. Therefore, incorporating the proposed LDA to other protocols may result in more gains, depending on the selection of the AIMD parameters.

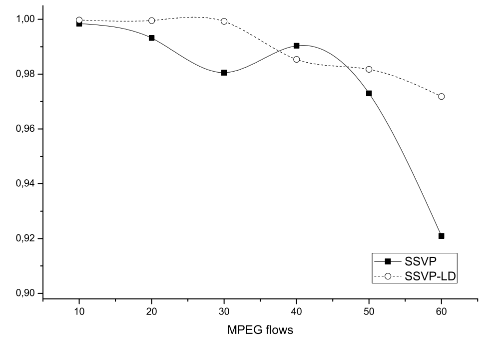
As depicted in Fig. 3b, SSVP-LA excels in bandwidths sharing, while SSVP also achieves satisfactory levels of fairness. Although SSVP's congestion control maintains adequate AIMD oscillation allowing all the system flows to converge to the fairness point, congestion along with wireless loss can undermine long-term fairness (i.e. 60 SSVP flows). In this case, the presence of the LDA notably improves fairness, as well as system stability.

In order to quantify the smoothness observed by the end-user, we trace the proportion of delayed packets, effectively capturing the effect of jitter. In [13] we showed that SSVP maintains a smooth sending rate in accordance with the QoS provisions of video streaming applications. SSVP's smoothness is also demonstrated in Fig. 3c, where the protocol achieves the timely delivery of most packets. However, the incorporated LDA further refines transmission rate fluctuations, abolishing the damage of error-induced multiplicative decrease on flow throughput and smoothness. As a result, SSVP-LD delivers a smoother video flow, especially when link errors are the primary cause for the observed packet drops (i.e. 10-40 flows).

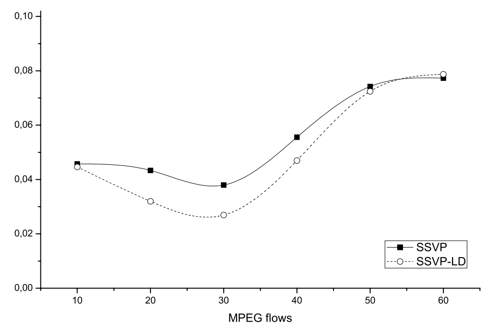
We also carried out the experiments using diverse error rate adjustments (PER: 0.02-0.08) in order to investigate the efficiency of the LDA in highly erroneous wireless links. We hereby demonstrate results from 30 flows (Fig. 4). Although packet error rates, as high as 0.08, are not common across wireless channels, such simulations provide useful insights into protocol sensitivity over links with high PER.




**(a)** System Throughput



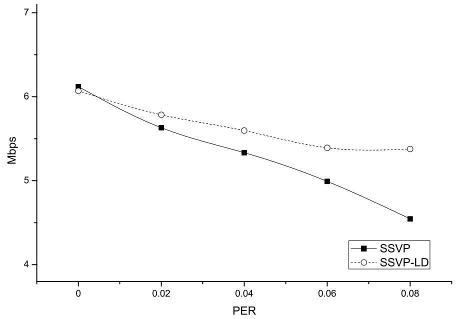

**(b)** Fairness Index



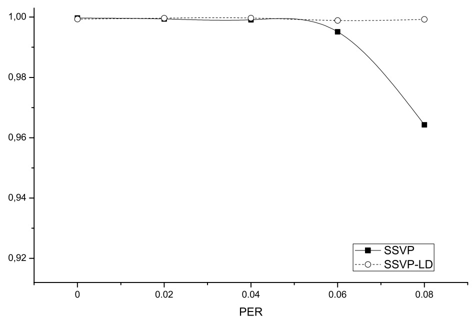

**(c)** Delayed Packets Rate

**Fig. 3.** Performance with wireless errors

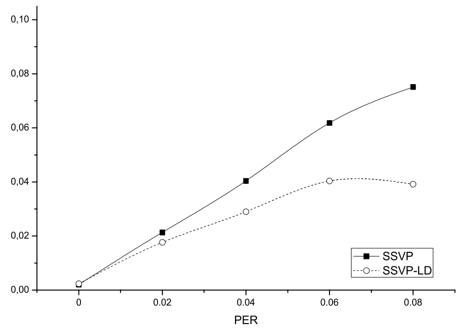
Fig. 4a illustrates that throughput performance notably degrades, especially in the case of SSVP. SSVP-LD is apparently less sensitive to the increased error rates, since the integrated LDA alleviates most of the undesirable effects induced by the link errors. We note that throughput degradation is sometimes inevitable, since wireless errors may occur while the queue length is close to the buffer size. Recall that we set *qthress* to 0.5. Before ending up to this adjustment, we enabled simulations with diverse *qthress* adjustments. Adjusting *qthress* at a higher value may slightly increase the throughput rate, but other undesirable implications are observed (i.e. smoothness might be compromised). Essentially,




**(a)** System Throughput

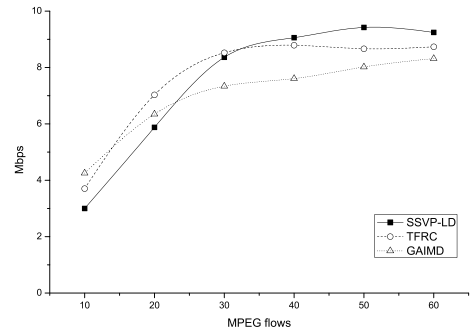



**(b)** Fairness Index

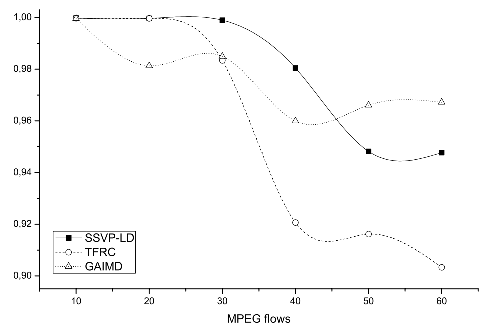



**(c)** Delayed Packets Rate

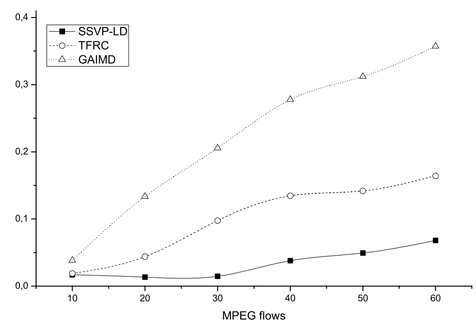
**Fig. 4.** Performance with diverse error rates




**(a)** Throughput of MPEG flows




**(b)** Fairness Index




**(c)** Delayed Packets Rate

**Fig. 5.** Performance with wireless errors and interfering traffic

an optimal setting for *qthress* is subject to the error characteristics of the underlying network, device constraints and performance trade-offs.

Apart from the throughput gains achieved by the LDA, Fig. 4b reveals that fairness for SSVP-LA is not compromised, even for error rates as high as 0.08. Errors are not concurrently experienced by all flows; hence, some flows may back off while the others may keep growing. This *partial* downward adjustment upon packet loss results in varying flow throughput rates and has a direct impact on fairness. This observation is primarily applicable to SSVP, as well as to other protocols that can not detect the nature of the error. SSVP-LD nearly avoids this implication, since multiplicative decrease is invoked only when the queue occupies a large proportion of the buffer size.

In terms of video delivery, delay variation becomes more evident, as the error rate increases (Fig. 4c). Packet errors occasionally induce interruptions in the sending rate and the perceptual video quality inevitably deteriorates. The proposed LDA manages to alleviate most of these impairments and sustain a relatively smooth video flow. On the contrary, SSVP's downward adjustments in response to the transient errors cause perceptible oscillations in the sending rate, with the effect of jitter becoming evident to the end-user. We observe that the rate of delayed packets increases almost in proportion to PER for SSVP. In contrast, SSVP-LD effectively enforces an upper bound to the magnitude of delay variation, providing a possible guarantee for streaming applications that can efficiently operate within this QoS provision.

We conclude our performance studies by evaluating SSVP-LD versus TFRC and GAIMD. Both TFRC and GAIMD are implied to yield remarkable efficiency on media delivery over a wide range of network and session dynamics. Along these lines, we simulated a diverse range of MPEG flows (10-60) over (i) SSVP-LD, (ii) TFRC, and (iii) GAIMD, competing with 5 FTP connections over TCP Reno, successively. PER is adjusted at 0.02. Similarly to the previous scenarios, we obtained the corresponding *Throughput*, *Fairness Index* and *Delayed Packets Rate* measurements (Fig. 5).

According to Fig. 5a, both SSVP-LD and TFRC utilize a high fraction of the available bandwidth. SSVP-LD remains relatively immunized by the wireless errors, as well as by the interfering FTP flows. The protocol apparently operates more efficiently during high link-multiplexing, since it performs downward adjustments primarily in response to congestion. TFRC also responds adequately to the link errors, estimating the loss rate and adjusting the sending rate approximately. On the contrary, GAIMD fails to adapt to the network dynamics, since it can not detect the nature of the error. The presence of both congestion and error-induced loss eventually diminishes the protocol's throughput rate.

Fig. 5b illustrates that SSVP-LD and GAIMD achieve high levels of fairness. The AIMD-based responses during congestion enforce competing flows to converge to the fairness point for both protocols. On the other hand, we observe that the *Fairness Index* for TFRC degrades abruptly, reflecting a throughput imbalance between the connections. Apparently, TFRC's equation-based responses to packet loss undermine long-term fairness, along with contention increase.

According to Fig. 5c, SSVP-LD achieves the timely delivery of video packets maintaining an uninterrupted and smooth sending rate that is slightly affected by transient errors and contention. We also observe that TFRC's random downward adjustments induce oscillations in the sending rate, and subsequently delay variation of a considerable magnitude. GAIMD's performance on video delivery may as well frustrate the end-user. In dynamic environments with transient errors, the protocol's congestion-oriented responses to all types of errors counterbalance the potential gains from a gentle decrease ratio (that could favor smoothness in a static and error-free network).

## V. Conclusions

We have relied on a combined approach to effectively support the delivery of streaming video over heterogeneous environments. In this context, we analyzed the interaction between a transport protocol with appropriate AIMD parameters and an end-to-end mechanism for loss differentiation. Simulation results have validated the feasibility and efficiency of our approach in terms of bandwidth utilization, video delivery, and fairness. According to our knowledge, SSVP-LD composes one of the few available end-to-end schemes that achieve efficient performance on video delivery in wired/wireless networks, without requiring the support from lower-layer feedback or AQM mechanisms.

## References


[1] H. Balakrishnan, V. Padmanabhan, S. Seshan, and R. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links", *ACM/IEEE Transactions on Networking*, 5(6), 1997, pp. 756-769.

[2] D. Bansal and H. Balakrishnan, "Binomial Congestion Control Algorithms", in *Proc. IEEE INFOCOM 2001*, Anchorage, Alaska, USA, April 2001.

[3] A. Capone, L. Fratta, and F. Martignon, "Bandwidth Estimation Schemes for TCP over Wireless Networks", *IEEE Transactions on Mobile Computing*, 3(2), 2004, pp. 129-143.

[4] S. Cen, P. C. Cosman, and G. M. Voelker, "End-to-end Differentiation of Congestion and Wireless Losses", *IEEE/ACM Transactions on Networking*, 11(5), October 2003, pp. 703-717.

[5] D. Chiu and R. Jain, "Analysis of the increase/decrease algorithms for congestion avoidance in computer networks", *Journal of Computer Networks and ISDN*, 17(1), June 1989, pp. 1-14.

[6] A. Chockalingam, M. Zorzi, and V. Tralli, "Wireless TCP performance with link layer FEC/ARQ", in *Proc. IEEE ICC '99*, Vancouver, Canada, June 1999.

[7] S. Floyd and K. Fall, "Promoting the use of end-to-end congestion control in the Internet", *IEEE/ACM Transactions on Networking*, 7(4), August 1999, pp. 458-472.

[8] S. Floyd, M. Handley, J. Padhye, and J. Widmer, "Equation-Based Congestion Control for Unicast Applications", in *Proc. ACM SIGCOMM 2000*, Stockholm, Sweden, August 2000.

[9] S. Floyd and V. Jacobson, "Random Early Detection gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, 1(4), August 1993, pp. 397-413.

[10] T. Goff, J. Moronski, D. Phatak, and V. Gupta, "Freeze-TCP: A true end-to-end Enhancement Mechanism for Mobile Environments", in *Proc. IEEE INFOCOM 2000*, Tel-Aviv, Israel, March 2000.

[11] F. Hu and N. K. Sharma, "Enhancing Wireless Internet Performance", *IEEE Communications Surveys and Tutorials*, 4(1), 2002, pp. 2-15.

[12] E. Kohler, M. Handley, and S. Floyd, "Designing DCCP: Congestion control without reliability", in *Proc. ACM SIGCOMM 2006*, Piza, Italy, September 2006.

[13] P. Papadimitriou and V. Tsaoussidis, "End-to-end Congestion Management for Real-Time Streaming Video over the Internet", in *Proc. 49$^{th}$ IEEE GLOBECOM*, San Francisco, USA, November 2006.

[14] P. Papadimitriou and V. Tsaoussidis, "On Transport Layer Mechanisms for Real-Time QoS", *Journal of Mobile Multimedia*, 1(4), January 2006, pp. 342-363.

[15] P. Papadimitriou and V. Tsaoussidis, "SSVP: A Congestion Control Scheme for Real-Time Video Streaming", *Technical Report: TR-DUTH-EE-2006-17*, Available at: http://utopia.duth.gr/~ppapadim/ssvp.pdf.

[16] K. Ramakrishnan and S. Floyd, "A proposal to add explicit congestion notification (ECN) to IP", *RFC 2481*, January 1999.

[17] R. Rejaie, M. Handley, and D. Estrin, "RAP: An End-to-end Rate-based Congestion Control Mechanism for Realtime Streams in the Internet", in *Proc. IEEE INFOCOM 1999*, New York, USA, March 1999.

[18] P. Sinha, N. Venkitaraman, R. Sivakumar, and V. Bharghavan, "WTCP: A Reliable Transport Protocol for Wireless Wide-Area Networks", in *Proc. ACM MOBICOM '99*, Seattle, USA, August 1999.

[19] V. Tsaoussidis and H. Badr, "TCP-Probing: Towards an Error Control Schema with Energy and Throughput Performance Gains", in *Proc. 8$^{th}$ IEEE Int/nal Conference on Network Protocols* (*ICNP*), Osaka, Japan, November 2000.

[20] V. Tsaoussidis and I. Matta, "Open issues on TCP for Mobile Computing", *Journal of Wireless Communications and Mobile Computing*, 2(1), February 2002, pp. 3-20.

[21] V. Tsaoussidis and C. Zhang, "The Dynamics of Responsiveness and Smoothness in Heterogeneous Networks", *IEEE Journal on Selected Areas in Communications* (*JSAC*), 23(6), June 2005, pp. 1178-1189.

[22] Y. R. Yang, M. S. Kim and S. S. Lam, "Transient Behaviors of TCP-friendly Congestion Control Protocols", in *Proc. IEEE INFOCOM 2001*, Anchorage, Alaska, USA, April 2001.

[23] Y. R. Yang and S. S. Lam, "General AIMD Congestion Control", in *Proc. 8$^{th}$ IEEE Int/nal Conference on Network Protocols (ICNP)*, Osaka, Japan, November 2000.